

HIGH PERFORMANCE ADAPTIVE FINITE IMPULSE RESPONSE FILTER USING NEW DISTRIBUTED ARITHMETIC ALGORITHM

SANDEEP KUMAR*, VIGNESWARAN T

Department of VLSI, School of Electronics Engineering (SENSE), VIT Chennai, Tamil Nadu, India. Email: erskumar.kumar@gmail.com

Received: 19 January 2017, Revised and Accepted: 20 February 2017

ABSTRACT

Finite impulse response (FIR) filters are very important in signal processing applications. This research is to analyze the performance of FIR filter with the Xilinx Software. The distributed arithmetic (DA) algorithm is extensively used in FIR filter to improve its speed and reducing the area of the filter. The design of low power filter will be achieved by pipelining and parallel processing concept on DA. The aim is to design a filter which has less delay time and supports the pipelining/parallel processing feature, helps in high speed with less power dissipation and area. The paper discusses field-programmable gate array implementation of FIR filter, and due to parallel data processing, its computation is fast and also provides an efficient architecture in terms of area and power consumption. New distributed arithmetic is a high performance and for low power filter.

Keywords: Finite impulse response, Distributed algorithm, Distributed arithmetic.

© 2017 The Authors. Published by Innovare Academic Sciences Pvt Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>) DOI: <http://dx.doi.org/10.22159/ajpcr.2017.v10s1.19754>

INTRODUCTION

The digital finite impulse response (FIR) filter is generally implemented in every sort of frequency response. FIR filter normally implemented in pulse shaping filter, digital phase-locked loop, as an anti-aliasing filter for sampling. FIR filter consists of many lookup table (LUTs) and multiply and accumulate block and shift accumulator. The distributed arithmetic (DA) gives us a multiplication-free method to calculate its inner product based on LUT. In LUT the partial product is precomputed and directly gives us output which makes computation fast. Recently, as demand is increasing of digital filter FIR filter are basic blocks for various digital signal processing (DSP). Many schemes have been applied to FIR filter to increase the efficiency of the filter in terms of speed and reducing the power consumption. New DA algorithm is also a part to increase the efficiency. FIR filters intensively used in all video and communication speed due to its various features DA has a feature of serial word nature. To represent 2's complement in digital circuits, it's easy to perform as arithmetic operations are easy to perform. But respective to that dynamic power consumption increases and also increases area because when the signal is processed in the digital circuit it cannot use the entire bit as it is shown in Hwang [1]. To avoid the increase in dynamic power consumption of digital circuit we can choose the sign-magnitude representation due to that it can be used in FIR filter to take less power consumption, but it results in increase in hardware complexity as positive and negative operations require different operations for both numbers and LUT is the basic block of field-programmable gate array (FPGA) which helps in generating Boolean function [2]. This literature review basically focused on FIR filter, to utilize the low power consumption, improving in efficiency and for better performance. Litwin find the concept of the digital filter in 2000 used in a discrete time and discrete amplitude convolver [3]. The filter used for this application is digital filter as they have a vast range of application in DSP system. Basically, there are two types of filter for this type of application, that is, FIR and infinite impulse response depending on the requirement of application. Chitra E in April 1971 [4] discussed about the techniques used for vast application including frequency sampling and window method. With the use of these techniques and their characteristic efficiency have been compared. At the time of comparing a lot of issues occur first solve that issues and they compare like difficulty with the Fourier coefficient and in calculating Bessel function. The transition bandwidth in every case can be finding using Kaiser Window. Basically, FIR filter can be designed using a lot of methods, but all of them are based on ideal filter

approximation. The idea is not to achieve ideal characteristics, as it is almost impossible to achieve the all characteristics of the filter. The FIR filter transfer function gives the idea that if the filter orders increases and results in increasing the complexity and the required time to samples the signal. The frequency response of oscillatory function or monotone function within a certain range. The waveform in output is based on the methods which have been used in the design process and the parameters which have been defined. Many papers describe the most popular techniques that are based on window function but on the other hand many algorithms are also there to improve the efficiency of FIR filter. The output characteristics of the transfer function deviates from the ideal frequency response. Both filters have the advantages and disadvantages based on the application choose the filter. In general, FIR filter has the linear phase characteristics. The architecture for implementing the low complexity and reconfigurable FIR filter was proposed by Vigneswaran *et al.* (2007). For various application, various methods have been implemented to improve the performance and efficiency. For e.g.: It can be used in channelizer, and the method or algorithm which has been implemented for the same is binary common subexpression elimination [5]. The adder has reduced using this algorithm. In this paper, new distributed algorithm has been used to improve the efficiency in terms of reducing the combinational path delay and increasing the speed. New distributed algorithm do not require different hardware for signed number as signed number is converted into an unsigned number using a method then it's implementation is same; This helps us in reducing the area and in increasing the speed. With respect to these our efficiency and performance in increased the results for the same as been discussed in below section.

CONVENTIONAL TECHNIQUE

In mathematical form, we can write FIR filter as

$$y[n] = \sum_{i=0}^{N-1} h[i]x[n-i] \quad (1)$$

Where x shows input signal, H is the filter coefficient, and y (n) is desired output signal. 4 tap filter mainly consists of delays and multipliers and summation and gets the result after alliteration [6]. Power and area can be calculated using Xilinx software and cadence tool is shown Fig. 1.

This paper implements multiplier block of FIR filter which helps in improving device utilization as DA throughput rate do not depend on filter length and data size. In 4 tap FIR filter 16-bit input is taken by input and some delay it goes to adder and this iteration goes on and finally get the output. In this algorithm, it does not require additional hardware for doing positive and negative numbers. The previous researchers show the reduction of adders up to 50% by implementing various algorithms.

DA first time implemented by Crosier [2] and used for signed numbers and proves the better result also. Table 1 showing a single term is shown for example.

Table 2 shows the output for each 4-bit output which represents by a single term in the table. On the other hand, we are reusing the LUT data to save the memory. After that, it has been implemented on FPGA design to save MAC blocks [7]. High speed using DA has been implemented in LMS architecture also.

Fig. 2 have some notations: A1, A2, A3, and A4 are adders and D is a delay. If $h(n)$ represents shows filter coefficient and $x[n]$ is the input signal to be processed, therefore DA can be shown in reference [2].

$$y = 2^B x_B[n] h[n] + \sum_{b=0}^{B-1} 2^b \sum_{N=0}^{N-1} h[n] x_b[n] \tag{2}$$

FIR filter generally consists of delays, multipliers and adder's which gives the filter output. MAC operation is intensively used in all DSP Algorithms. In this letter, the paper [1], has following equations and the letter d_j is one and given in reference Hwang [1].

$$Y = \left(\frac{1}{2} \sum_{j=0}^{w-1} \left(\sum_{i=0}^{N-1} c_i d_{i,w-1-j}\right) 2^{-j}\right) - \left(\frac{1}{2} \sum_{i=0}^{N-1} c_i 2^{-(w-1)}\right) \tag{3}$$

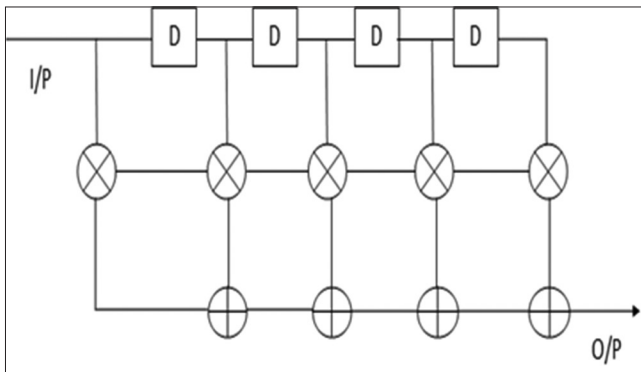


Fig. 1: 4-bit finite impulse response filter

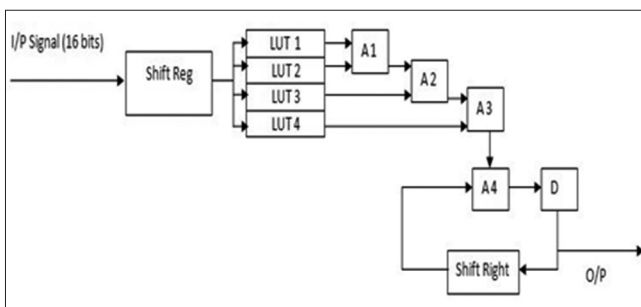


Fig. 2: Block diagram of conventional finite impulse response filter

$$= - \left(\sum_{j=0}^{W-1} \left(\sum_{i=0}^{N-1} |c_i| u_i \right) 2^{-j} \right) + \frac{1}{2} \left(\sum_{j=0}^{W-1} 2^{-j} \sum_{i=0}^{N-1} |c_i| - \sum_{i=0}^{N-1} |c_i| - \sum_{i=0}^{N-1} c_i 2^{-(W-1)} \right) \tag{4}$$

Where, u_i^j represents the sign of input sequence which can be positive it can be negative, d_i represents 2's compliment and c is coefficient and the $w-1$ represents bit word with the help of this LUT generates unsigned number also [8]. It helps in saving area and results in good speed also. The above equation the term inside the brackets represents a binary and operation involving the bit for input variable and bits for constant. It requires the arithmetic addition operation for the plus sign. The scaled parts in bracketed represent the exponential factor. The every product term contains a variable (signal) and a constant (coefficient which is in fixed point binary format but do not depend on the word length: It computes the product of the terms by turn basis and the partial products for the same are computed and stored differently and it takes the bit by bit to compute the partial product. Normally, these partial products are also the coefficients. Finally, all the partial products are added and results are produced.

PROPOSED WORK

The scheme implemented on FPGA using LUT-DA generally uses three main components: Shift/accumulator unit, one input register, and the main LUT unit. In addition, it also requires an adder to update partial product and multiplexer. The proposed technique requires less area and the maximum combinational path delay is 7.508 ns.

Table 1: 4-bit LUT

a_3	a_2	a_1	a_0	LUT DATA
0	0	0	0	0
0	0	0	1	b (0)
0	0	1	0	b (1)
0	0	1	1	b (0)+b (1)
0	1	0	0	b (2)
0	1	0	1	b (2)+b (0)
0	1	1	0	b (2)+b (1)
0	1	1	1	b (2)+b (1)+b (0)
1	0	0	0	b (3)
1	0	0	1	b (3)+b (0)
1	0	1	0	b (3)+b (1)
1	0	1	1	b (3)+b (1)+b (0)
1	1	0	0	b (3)+b (2)
1	1	0	1	b (3)+b (2)+b (0)
1	1	1	0	b (3)+b (2)+b (1)
1	1	1	1	b (3)+b (2)+b (1)+b (0)

Table 2: 4-bit LUT showing output term

a_3	a_2	a_1	a_0	Output term
0	0	0	0	h (0)
0	0	0	1	h (1)
0	0	1	0	h (2)
0	0	1	1	h (3)
0	1	0	0	h (4)
0	1	0	1	h (5)
0	1	1	0	h (6)
0	1	1	1	h (7)
1	0	0	0	h (8)
1	0	0	1	h (9)
1	0	1	0	h (10)
1	0	1	1	h (11)
1	1	0	0	h (12)
1	1	0	1	h (13)
1	1	1	0	h (14)
1	1	1	1	h (15)

Table 3: Design summary of FIR filter

Logic	Used	Available	Utilization
Number of Slices	3	960	0
Number of 4 input LUTs	5	1920	0
Number of I/P LUTs	2	4896	1
Number of bonded IOBs	17	66	25

FIR: Finite impulse response

Table 4: Gate used

Type	Instances	Area	Area (%)
Sequential	16	1134.302	16.0
Inverter	48	359.251	5.1
Buffer	11	146.362	2.1
Unresolved	3	0.000	0.0
Logic	238	5445.317	76.9
Total	316	7085.232	100

RESULTS AND DISCUSSION

The results and Table 3 is shown has been implemented in XILINX integrated software environment. The maximum combinational path delay of work is 7.508 ns. The Table 3 shows the number of slices used and a number of 4 input LUTs. Table 3 shows the power, area, and gates calculation using the cadence tool.

Table 3 shows the design summary of a 4 tap filters which is generated using XILINX software.

Table 4 shows the power consumption results using cadence tool. It shows the difference between leakage power and dynamic power. It gives the exact information about the power.

Table 5 shows the total number of the area used by each module using cadence tool and the tmsc library. In this, the area taken by each module is different in terms of cell and cell area (Table 6).

Table 5: Area used

Instance	Cell	Cell area	Total area	Library
fir_filter	313	7085	7085	tmsc18
add_28_26	17	1074	1074	tmsc18
add_27_26	21	875	875	tmsc18
mul_21_21	11	146	146	tmsc18

Table 6: Gate used

Type	Instances	Area	Area%
Sequential	16	1134.302	16.0
Inverter	48	359.251	5.1
Buffer	11	146.362	2.1
Unresolved	3	0.000	0.0
Logic	238	5445.317	76.9
Total	316	7085.232	100

REFERENCES

1. Hwang S. New distributed arithmetic algorithm for low - power FIR filter implementation. *IEEE Signal Proc Lett* 2004;11:463-6.
2. Nandal A, Vigneswaran T, Rana AK. An efficient 256-tap parallel fir digital filter implementation using distributed arithmetic architecture. 11th International Multi-Conference on Information Processing; 2015.
3. Khan S, Jaffery ZA. Low Power FIR implementation on FPGA using Parallel Distributed Arithmetic. *IEEE INDICON*; 2015.
4. Chitra E, Vigneswaran T. An efficient low power and high speed distributed arithmetic design for FIR filter. *Indian J* 2016;9(4):1-5.
5. Mohanty BK. A high-performance energy-efficient architecture for FIR adaptive filter based on new distributed arithmetic formulation of block LMS algorithm. *IEEE Trans Signal Proc* 2013;61(4):921-32.
6. White SA. Applications of distributed arithmetic to digital signal processing: Tutorial review. *IEEE ASSP* 1989;6:4-19.
7. Parhi KK. *VLSI Digital Signal Processing Systems: Design and Implementation*. New York: Wiley-Blackwell; 1999.
8. Vigneswaran T, Reddy PS. Design of digital fir filter based on dynamic distributed arithmetic algorithm. *Indian J Appl Sci* 2007;7(9):2908-10.